Information Theoretic Pairwise Clustering

Avishay Friedman and Jacob Goldberger

Engineering Faculty, Bar-Ilan University, Ramat-Gan 52299, Israel

Abstract. In this paper we develop an information-theoretic approach for pairwise clustering. The Laplacian of the pairwise similarity matrix can be used to define a Markov random walk on the data points. This view forms a probabilistic interpretation of spectral clustering methods. We utilize this probabilistic model to define a novel clustering cost function that is based on maximizing the mutual information between consecutively visited clusters of states of the Markov chain defined by the graph Laplacian matrix. The algorithm complexity is linear on sparse graphs. The improved performance and the reduced computational complexity of the proposed algorithm are demonstrated on several standard datasets.

1 Introduction

Effective automatic grouping of objects into clusters is one of the fundamental problems in machine learning and in other fields of study. In many approaches, the first step toward clustering a dataset is extracting a feature vector from each object. This reduces the problem to the aggregation of groups of vectors in a feature space. A commonly used algorithm in this case is the k-means. However, in many cases we are only given pairwise similarity information between data points. For example, in social networks, only binary neighborhood relations are given. In these cases k-means cannot be applied in a straightforward way. Instead, we seek for a partition of the data based on the similarity measure between the points. Out of the numerous clustering algorithms, spectral clustering [14,16] has gained considerable attention in recent years due to its strong performance on arbitrary shaped clusters, and its well-defined mathematical framework [20].

Another family of clustering algorithms, that are derived from information-theory concepts, corresponds to the case of distributional clustering. Here each data point is described as a distribution. This situation is illustrated by the generic example of document clustering based on word histograms [18],[17]. In this case, the mutual information between word occurrences and clusters of documents is a natural clustering criterion [19] [4] that has been proven to be powerful in many cases. The information-theoretical principle described above is only applicable when a feature distribution, associated with each data point, is provided as part of the problem setup. In this paper we extend the mutual information clustering criterion to the domain of pairwise clustering. The probabilistic interpretation of spectral clustering, based on a Markov random walk, is used to associate a distribution with each data point via the corresponding

E. Hancock and M. Pelillo (Eds.): SIMBAD 2013, LNCS 7953, pp. 106–119, 2013.

conditional distribution row in the Markov transition matrix. In particular, we define a random walk on the data points and maximize the mutual information between cluster labels of data-points that are visited during the random walk. We show that this results in an efficient clustering method with state-of-the-art performance on real-world datasets. The remainder of this paper is organized as follows. Section 2 defines the notation of similarity graphs and the associated Laplacian matrix. Section 3 describes the minimum information-loss criterion for clustering the Markovian random-walk states. Section 4 introduces the Information-Theoretic Pairwise Clustering (ITPC) algorithm. Section 5 reviews related work and Section 6 describes numerical experiments on several standard datasets.

2 Similarity Graphs and Random Walks

Given a set of data points $x_1, ..., x_n$ and some symmetric notion of similarity $w_{ij} \geq 0$ between all pairs of data points x_i and x_j, the goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. In the common case where the data points live in the Euclidean space R^d, a reasonable candidate for a similarity measure is the Gaussian function $w_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$ (where the parameter σ controls the width of the local neighborhoods). Ultimately, the choice of the similarity function depends on the domain the data come from and the specific clustering task.

In the case where we have information in the form of similarities between data points, we can represent the data as a similarity graph $G = (V, E)$. Each vertex i in this graph represents a data point x_i. Two vertices are connected if the similarity w_{ij} between the corresponding data points x_i and x_j is positive and the edge is weighted by w_{ij}. The problem of clustering can now be reformulated using the similarity graph: we want to find a partition of the graph in which existing edges between different groups have low weights and edges within a group have high weights.

Denote the similarity matrix by $W = (w_{ij})$. The degree of a vertex $i \in V$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The degree matrix D is defined as the diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal. The normalized Laplacian matrix L is defined as $L = I - D^{-1}W$ [1]. (Note that in the literature there is no unique convention as to which matrix exactly is called "Graph Laplacian" [20].) All variants of the spectral clustering algorithm are based on using eigenvectors of the Laplacian matrix to represent the abstract data points as points in the Euclidean space. The clusters can be then obtained by applying simple clustering algorithms such as k-means in the embedded space [14,16,22]. The matrix $P = D^{-1}W = I - L$ is a stochastic matrix (non-negative entries, row sums are all 1). Using $n \times n$ transition matrix P we can define a stationary Markov chain that corresponds to a random walk on the graph nodes. Let $X = \{X_t\}$

be the n-valued stationary Markov chain defined by:

$$P_{ij} = (D^{-1}W)_{ij} = p(X_2 = j | X_1 = i) = \frac{w_{ij}}{\sum_k w_{ik}} \tag{1}$$

The transition probability P_{ij} of jumping in one step from i to j is proportional to the edge weight w_{ij}. Let $\pi = (\pi_1, ..., \pi_n)$, where $\pi_i = d_i / (\sum_j d_j)$. It can be easily verified that $P^\top \pi = \pi$. Hence, if the graph is connected and non-bipartite, then π is the unique stationary distribution of the Markov chain defined by P [20]. Therefore, the joint stationary probability of X_1 and X_2 is:

$$p(X_1 = i, X_2 = j) = \frac{w_{ij}}{\sum_{kl} w_{kl}}. \tag{2}$$

Given the random walk model (1) we can translate the pairwise clustering problem, into the problem of clustering the states of a Markov chain.

3 Clustering the States of a Markov Chain

Let $\{A_1, ..., A_m\}$ be a partition of the states $\{1, ..., n\}$ into m clusters and let C denote the subset membership function, i.e. $C(i) = j$ if $i \in A_j$. For each t we define a random variable $Y_t = C(X_t)$ indicating the cluster membership of the state visited by the random walk at time t. The joint distribution of the random variables (Y_1, Y_2) defined on the clusters is:

$$\begin{aligned} p(Y_1 = i, Y_2 = j) &= p(X_1 \in A_i, X_2 \in A_j) \qquad (3) \\ &= \frac{1}{\text{vol}(V)} \sum_{k \in A_i, l \in A_j} w_{kl} \end{aligned}$$

such that $\text{vol}(V) = \sum_{ij} w_{ij}$. The model is illustrated by the following diagram:

$$\begin{tikzcd}
X_1 \arrow[r] \arrow[d, "C"'] & X_2 \arrow[r] \arrow[d, "C"'] & X_3 \arrow[r] \arrow[d, "C"'] & ... \arrow[d] \\
Y_1 & Y_2 & Y_3 & ...
\end{tikzcd}$$

Each clustering $\{A_1, ..., A_m\}$ induces a joint distribution $p(Y_1, Y_2)$ on the clusters visited on consecutive time units. To find the best clustering based on the joint distribution of Y_1 and Y_2, we need to extract from the $m \times m$ matrix $(p(Y_1 = i, Y_2 = j))$ a single number that measures the clustering quality. Once decided on a suitable clustering score, we can find the clustering that optimizes this score.

An intuitive clustering score, that we would like to minimize, is:

$$p(Y_2 \neq Y_1) = \sum_{i=1}^{m} p(Y_2 \neq i | Y_1 = i) p(Y_1 = i) \tag{4}$$

which is the probability that consecutive visited points would be in different clusters. However, the clustering that minimizes criterion (4) is the one formed by a single cluster that contains all the data points. Even if we enforce that all the m clusters should be non-empty, the score (4) still favors clusterings that are very unbalanced. To overcome this degeneracy we can modify the clustering score we minimize as follows:

$$\text{Ncut}(A_1, ..., A_m) = \sum_{i=1}^{m} p(Y_2 \neq i | Y_1 = i) \tag{5}$$

where Ncut is the Normalized-Cut score of the partition $C = \{A_1, ..., A_m\}$. Meila and Shi [12] showed that the Ncut spectral clustering algorithm [16] [21] is an algorithm that finds an optimal solution for a relaxation of the Ncut criterion (5) for clustering the states of the random walk defined by the Laplacian of affinity graph. Dhillon et el. [3] applied kernel k-means (with kernel $K = D^{-1}WD^{-1}$) to directly optimize the Ncut score.

In this study we suggest to apply the information-theoretical principle of minimal information loss to cluster the states of the random walk. The mutual information induced by the clustering $C = \{A_1, ..., A_m\}$ is:

$$MI(A_1, ..., A_m) = I(Y_1; Y_2) = \tag{6}$$
$$\sum_{ij} p(Y_1 = i, Y_2 = j) \log \frac{p(Y_1 = i, Y_2 = j)}{p(Y_1 = i)p(Y_2 = j)}.$$

The original walk over the points also determines a walk over the clusters. The goal of clustering is to choose the clustering such that the loss in mutual information due to clustering is minimized. A good Markov-state clustering should preserve maximum information on the visited points. Using the mutual information criterion, the best clustering of the given n points into m clusters is the one that minimizes the information loss of the mutual information $I(X_1; X_2) - I(Y_1; Y_2)$ over all the partitions of the state-space into m subsets. The definition of mutual information implies that:

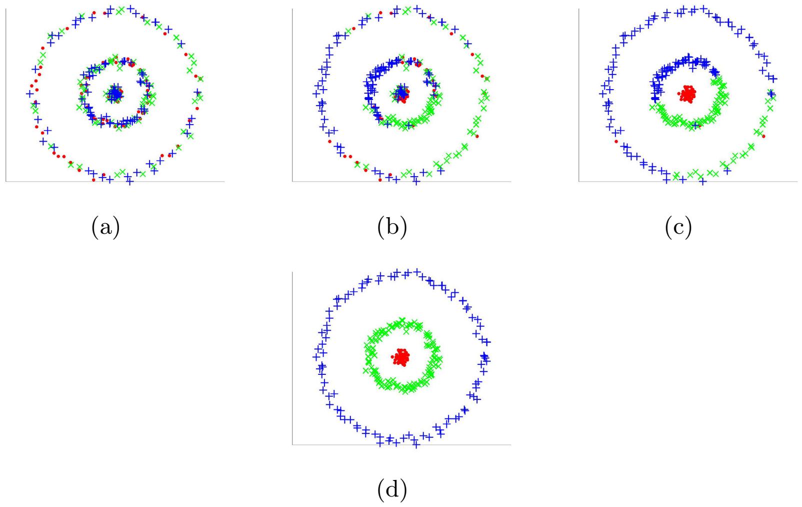
$$I(Y_1; Y_2) = H(Y_2) - \sum_{i=1}^{m} H(Y_2 | Y_1 = i)p(Y_1 = i) \tag{7}$$

When maximizing $I(Y_1; Y_2)$ the first term of (7) encourages clusters to have similar sizes and the second term discourages the random walk from jumping from cluster to cluster.

Utilizing standard information-theory manipulations we can derive several equivalent forms for the information loss function we want to minimize.

$$\begin{aligned} &\text{score}(C) = I(X_1; X_2) - I(Y_1; Y_2) \qquad (8)\\ &= D(p(X_1, X_2) \| p(Y_1, Y_2)p(X_1|Y_1)p(X_2|Y_2))\\ &= H(Y_1, Y_2) + H(X_1|Y_1) + H(X_2|Y_2) - H(X_1, X_2)\\ &= D(p(X_2|X_1) \| p(X_2|Y_1)) + D(p(Y_1|X_2) \| p(Y_1|Y_2)) \end{aligned}$$

where $Y_1 = C(X_1)$, $Y_2 = C(X_2)$, D is the Kullback-Leibler divergence and H is the entropy function [2]. The optimal state-clustering is the one that minimizes the information-loss function score(C). Note that the information-theoretic equality (8) is correct for clustering the states of a general Markov chain. In our case, because the similarity matrix is symmetric, the Markov chain is also reversible.



(a)

(b)

(c)

(d)

Fig. 1. The steps of the ITPC algorithm on a three-circles data set. (a) random initializing, (b),(c) intermediate results, (d) final results (obtained after two passes over the data points).

Following [14], to understand the cost function we optimize, it is instructive to consider its behavior in the "ideal" case in which all points in different clusters are infinitely far apart and the m clusters are equal in shape. In this case the joint cluster distribution $(C(X_1), C(X_2))$ of the correct clustering is the $m \times m$ scalar matrix $\frac{1}{m}I$. Hence, for the correct clustering $H(C(X_1)) = \log(m)$ and $H(C(X_2)|C(X_1)) = 0$ and therefore, $I(C(X_1); C(X_2)) = \log(m)$. However, for any joint distribution (U, V) on $m \times m$ elements we have: $I(U; V) = H(U) - H(U|V) \leq H(U) \leq \log(m)$. Hence, the mutual information score $I(C(X_1); C(X_2))$ of the correct clustering is maximal.

4 The Clustering Algorithm

There is no closed-form solution for the minimal information-loss criterion stated in the previous section. Several standard optimization algorithms can be utilized to find the best clustering. In this study we apply a greedy sequential algorithm (see e.g. [17]). The sequential greedy algorithm has been found to perform well in terms of both clustering quality and computational complexity. The sequential clustering algorithm starts with a random clustering of the n graph nodes into m clusters. We then go over the data points in a circular manner and check for each point whether its removal from one cluster to another can reduce the information loss. This loop is iterated until no single-point transition offers an improvement. Since there is no guarantee that the algorithm will find the global optimum, we can run the algorithm on several initial random partitions and choose the best local optimum. Alternatively we can use a multi-level clustering approach [9].

The basic step of this algorithm is computing the information loss caused by merging a singleton cluster into an existing cluster. More generally we can define a distance measure between two clusters as the information-loss caused by merging the two clusters into a single one; i.e. the difference between the mutual information before and after the two clusters are merged. Direct computation of $I(Y_1;Y_2)$ requires $O(m^2)$ operations where m is the number of clusters. We next show that we can efficiently compute the information loss caused by merging two clusters in a time that is linear in the number of clusters.

Assume we are given a data partition $\{A_1, ..., A_m\}$ and we want to compute the information loss caused by merging the clusters A_1 and A_2 to obtain a new partition $\{A_1 \cup A_2, A_3, ..., A_m\}$ composed of $m-1$ clusters. Let Y_1 and Y_2 be the cluster membership random variable associated with the original clustering $\{A_1, ..., A_m\}$ and $\hat{Y}_1$ and $\hat{Y}_2$ are the cluster membership random variables associated with the clustering after merging A_1 and A_2 into a single cluster. The following formula provides an efficiently computed expression for the information loss caused by the merging:

$$d(A_1, A_2) = I(Y_1;Y_2) - I(\hat{Y}_1;\hat{Y}_2) \tag{9}$$

$$= 2\sum_{i=1}^{2}\sum_{j=1}^{m} p(Y_1=i, Y_2=j) \log \frac{p(Y_2=j|Y_1=i)}{p(Y_2=j|Y_1 \in \{1,2\})}$$

$$-\sum_{i=1}^{2}\sum_{j=1}^{2} p(Y_1=i, Y_2=j) \log \frac{p(Y_2=j|Y_1=i)}{p(Y_2 \in \{1,2\}|Y_1 \in \{1,2\})}$$

$$= 2p(Y_1 \in 12) JS(p(Y_2|Y_1=1)||p(Y_2|Y_1=2))$$

$$-p(Y_1 \in 12, Y_2 \in 12) I(Y_1;Y_2|Y_1 \in 12, Y_2 \in 12)$$

such that JS is the Jensen-Shannon divergence [2] and '12' is an abbreviation for $\{1,2\}$. The equality follows from the fact that the joint distributions of (Y_1, Y_2) and $(\hat{Y}_1;\hat{Y}_2)$ are very similar. For every i, j that are both larger than 2 we have $p(Y_1 = i, Y_2 = j) = p(\hat{Y}_1 = i, \hat{Y}_2 = j)$. Hence, most terms in the difference $I(Y_1;Y_2) - I(\hat{Y}_1;\hat{Y}_2)$ are canceled and the distance measure $d(A_i, A_j)$ (9) can be computed in $O(m)$ operations where m is the number of clusters. The sequential clustering algorithm requires the computation of the change in the cost function when moving a point from one cluster to another. This can be efficiently done using expression (9).

Table 1. The Information-Theoretic Pairwise Clustering (ITPC) algorithm

```
Input: A similarity graph defined by the $n \times n$ weight matrix W.
Output: A partition of the graph vertices into $m$ clusters.

Algorithm:

  1. Convert the graph into a Markov chain:

       $\widetilde{w}_{ij} \triangleq p(X_1 = i, X_2 = j) = \frac{w_{ij}}{\sum_{kl} w_{kl}}$

  2. Choose a random partition $A_1, ..., A_m$ of the Markov states and compute the
     cluster distribution $m \times m$ matrix:

       $q_{ij} = p(Y_1 = i, Y_2 = j) = p(X_1 \in A_i, X_2 \in A_j).$

  3. Loop until there is no change
       – for $i = 1, ..., n$ move state $i$ into the cluster that minimizes the information
         loss.
           • Remove state $i$ from its current cluster.
           • for $j = 1, ..., m$
               ∗ Add state $i$ to cluster $A_j$ and compute $d(\{i\}, A_j)$ (see Eq. (9)).
           • Choose the cluster which minimize the information-loss.

Removing/Adding state $i$ from/to cluster $A_j$ in a constant time (assuming
each node has at most $k$ neighbors):

  – Go over all $s \in$neighbors of node $i$
      • Assume $s$ is in cluster $A_l$.
      • $q_{jl} \leftarrow q_{jl} - \widetilde{w}_{is}$   /   $q_{jl} \leftarrow q_{jl} + \widetilde{w}_{is}$
      • $q_{lj} \leftarrow q_{lj} - \widetilde{w}_{is}$   /   $q_{lj} \leftarrow q_{lj} + \widetilde{w}_{is}$
```

The computational complexity of the proposed clustering algorithm is as follows. To recompute the joint distribution of (Y_1, Y_2) after moving a point i from one cluster to another we need to go over all weights on edges connected to i. Hence, it takes $O(n)$ for the basic step of searching all possible cluster memberships of a given data point. Assuming a fixed number of iterations over the dataset, the complexity is $O(n^2)$. In the (usual) case where the graph is sparse and each point is connected to at most k neighbors, the number of operations needed to recompute the clustering joint distribution, after moving a point from one cluster to another, is bounded by k. Hence, the computational complexity for sparse graphs is linear in the size of the dataset n. Note that when using spectral clustering methods, finding the eigenvectors of a large matrix is computationally costly. It takes $O(n^3)$ in general, and even with fast approximating techniques vast amount of space and time are required for larger datasets. We dub the proposed algorithm "Information-Theoretic Pairwise Clustering" (ITPC).

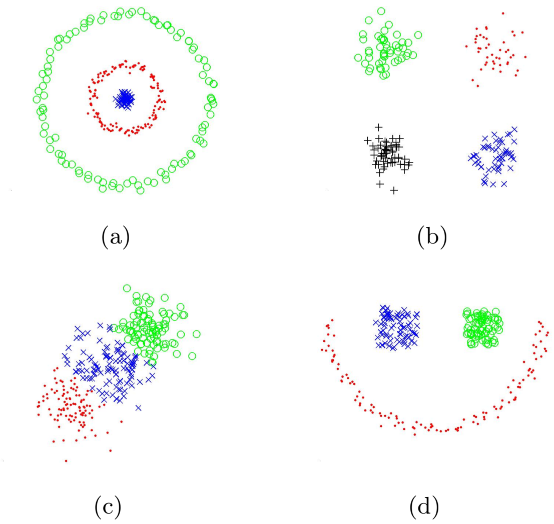


Fig. 2. Clustering of several synthetic datasets by ITPC (using Euclidean knn graph)

The linear time implementation of the ITPC algorithm is summarized in Table 1. An example of applying the sequential procedure on a synthetic dataset is shown in Figure 1.

One drawback of the sequential algorithm (in contrast to agglomerative approaches) is that the number of clusters must be given as input to the algorithm. In case we do not know the exact number of clusters we can slightly modify the algorithm in such a way that we can simply provide a rough estimation (upper bound) on the number of desired clusters. Consider the case of a cluster that contains a single object i. The iterative-sequential algorithm will not merge i into any other cluster because obviously this cannot increase the cost function $I(Y_1; Y_2)$. The algorithm will always prefer to leave i as a single member of a cluster. In the modified version we enforce a singleton cluster to be merged into another cluster. More generally if a cluster size is less than a predefined number, we enforce the cluster's members to be merged into other clusters. This step reduces the number of clusters by one. Utilizing this scheme, the number of output clusters can be adapted to the data.

5 Related Work

Information-theoretic approaches have been intensively used for data clustering algorithms. The standard problem setup is based on a given joint distribution of objects and features denoted by the random variables X_1 and X_2 respectively.

A one-sided clustering of the object set X_1, denoted by $C(X_1)$, aims to maximize the mutual information $I(C(X_1); X_2)$ between the object clusters and the features [19,17]. A co-clustering (aka two-sided clustering) applies a clustering procedure on both the objects set and the feature set. Denote the object clustering by $C_1(X_1)$ and the feature clustering by $C_2(X_2)$. The best co-clustering is the one that maximizes the mutual information between the object clusters and the feature clusters $I(C_1(X_1); C_2(X_2))$ [5]. Note that in the co-clustering setup the object set and the feature set are different and therefore the object clustering and the feature clustering are different. In our setup of pairwise clustering the objects set and the feature set are the same and therefore by clustering the objects we automatically also cluster the features. The two random variables X_1 and X_2 correspond to two instances of the same set and the *same* clustering function is *simultaneously* applied to the two random variables X_1 and X_2. The target is to find a clustering C such that the mutual information $I(C(X_1); C(X_2))$ is maximized. The three clustering cases are illustrated bellow:

$$Y_1 \xleftarrow{C} X_1 \longleftrightarrow X_2 \qquad \text{one-sided} \tag{10}$$

$$Y_1 \xleftarrow{C_1} X_1 \longleftrightarrow X_2 \xrightarrow{C_2} Y_2 \qquad \text{two-sided} \tag{11}$$

$$Y_1 \xleftarrow{C} X_1 \longleftrightarrow X_2 \xrightarrow{C} Y_2 \qquad \text{simultaneous} \tag{12}$$

Sequential optimization algorithm has been applied for one-sided clustering [17]. In that case if the number of features is kept fixed, the algorithm is linear in the number of data points. The basic step of the sequential algorithms is finding the best cluster assignment for a given point. This step requires computing the Jensen-Shannon (JS) divergence between the cluster and the point. Computing the JS divergence is linear in the number of features. Hence, in our case of pairwise clustering, where the number of features is equal to the number of data points, the complexity of the one-sided algorithm is quadratic in the data size. Note that even if the graph is sparse and therefore the distribution corresponds to each object is sparse, the cluster distribution is not necessarily sparse. Hence, the complexity of the one-sided clustering algorithm [17], applied to pairwise clustering problem, is quadratic. When applying a sequential optimization to the case of co-clustering (11), we need to iterate between feature clustering given the object clusters and object clustering given the feature clusters [5]. In contrast to previous methods, the complexity of the proposed ITPC algorithm when applied to sparse pairwise clustering is linear and there is no need to iterate between feature clustering and object clustering. An information theoretic clustering approach of the states of a general Markov chain has been suggest in [7]. Unlike our algorithm whose complexity is linear (on sparse graphs), the complexity of their algorithm is quadratic in the dataset size. Another iterative bipartition algorithm which uses JS divergence as the statistical dissimilarity measure has been suggest in [6].

Spectral clustering algorithms [14] [16], are based on finding a low dimensional embedding using eigenvector computation which can be slow. The Power Iteration Clustering (PIC) [10] is a variant of spectral clustering that directly finds the low-dimensional embedding. Graclus [3] is another efficient graph clustering algorithm that is based on directly optimize the Ncut score using multilevel kernel k-means and avoids the eigenvector computations. The main difference between our algorithm and the Graclus algorithm [3] is the cost function that is being optimized. We optimize the mutual information score (6) while Graclus optimizes the Ncut score (5). Another minor difference is that Graclus uses a batch version of the kernel k-means which is not guaranteed to converge if the kernel is not positive definite. We use a sequential greedy algorithm which monotonically improves the cost function and therefore always converges to a local optimum.

6 Experimental Results

In this section, we demonstrate our proposed ITPC method on the following commonly used real-world datasets: **Iris** contains flower petal and sepal measurements from three species of irises, 150 instances. **Glass** has 214 instances separated into six classes of glass. **Wine** are the results of a chemical analysis of wines. The analysis determined the quantities of 13 constituents found in each of three types of wines. 178 instances. **Wisconsin Diagnostic Breast Cancer (WDPC)** has 359 instances separated into two classes. Each instance has 30 continuous features. Features are computed from a digitized image of a fine needle aspiration (FNA) of a breast mass. **Olivetti Faces (OlFace5)** 10 images of 5 different people, 64 × 64 size [15]. **USPS-01:** 1100 instances of handwritten digits 0 and 1 from the USPS dataset. **USPS-17:** 1100 instances of handwritten digits 1 and 7 from the USPS dataset. **USPS-245:** 1650 instances of handwritten digits 2,4 and 5 from the USPS dataset [8]. **20ng*** are subsets of the 20 newsgroups text dataset [13]. The dataset 20ngA contains 100 documents from 2 newsgroups: misc.forsale and soc.religion.christian, 20ngB adds 100 documents to each group of 20ngA, 20ngC adds 200 from talk.politics.guns to 20ngB and 20ngD adds 200 from rec.sport.baseball to 20ngC.

To construct the pairwise similarity matrix we first need to choose a kernel and tune its parameters. Automatic parameter and kernel selection for unsupervised learning is still a difficult problem. Furthermore, different parameter values may be found to be optimal for different clustering algorithms. To avoid this problem we chose parameter-free affinity matrices. For the text datasets **20ng***, the affinity matrix we used is the cosine similarity between feature vectors. Note that no parameter needs to be tuned in the cosine kernel. In all other datasets, we used the k-nearest neighbor graph, based on the Euclidean distance, to construct the pairwise relations. We set $w_{ij} = 1$ if node i is a k-nearest neighbor of node j or j is a k-nearest neighbor of i. Otherwise, we set $w_{ij} = 0$.

Table 2. Clustering performance comparison on several real datasets. For all measures a higher number means better clustering. Bold numbers mark the best results for each dataset.

		PIC			**NJW**			**Graclus**			**ITPC**		
		[10]			[14]			[3]					
Dataset	**k**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**	**Pur**	**NMI**	**RI**
Iris	3	.687	.510	.727	.900	.778	.887	.840	.722	.837	**.973**	**.901**	**.966**
Glass	6	.579	.235	.702	.594	.299	.718	.584	.280	.713	**.626**	**.326**	**.727**
Wine	3	.961	.837	.947	**.966**	**.878**	**.955**	**.966**	**.878**	**.955**	.955	.847	.940
WDBC	2	.747	.307	.622	.932	.628	.872	**.947**	**.719**	**.900**	.893	.494	.809
OlFace5	5	.500	.365	.754	.560	.439	.793	.600	**.462**	**.806**	**.620**	.460	.803
USPS-01	2	.692	.243	.574	.988	.915	.982	**.991**	**.934**	**.982**	**.991**	**.934**	**.982**
USPS-17	2	.548	.010	.505	.979	.856	.959	**.982**	**.869**	**.964**	**.982**	**.869**	**.964**
USPS-245	3	.700	.510	.765	.664	.492	.707	.864	.660	.847	**.958**	**.844**	**.947**
20ngA	2	**.960**	**.759**	**.923**	**.960**	**.759**	**.923**	.945	.701	.896	.955	.736	.914
20ngB	2	.885	.568	.796	.508	.030	.500	.927	.626	.865	**.958**	**.747**	**.919**
20ngC	3	.642	.489	.692	.625	.339	.679	.603	.387	.678	**.713**	**.401**	**.736**
20ngD	4	.539	.295	.650	.504	.281	.669	.599	**.402**	.687	**.616**	.345	**.748**
Average		.703	.427	.721	.765	.558	.803	.821	.637	.844	**.853**	**.659**	**.871**

To evaluate the performance of the clustering methods we measured the clustering results against the true labels using three external validation indices: cluster purity (Pur), normalized mutual information (NMI), and the Rand index (RI). We used all these measures to ensure a more thorough evaluation of clustering results due to the different characteristics of each measure. We refer the reader to [11] for details regarding these measures.

Table 2 presents the results of comparing ITPC to three other clustering algorithms: Spectral clustering (NJW) [14], Power Iteration Clustering (PIC) [10] and the Graclus algorithm [3]. We also tried the Ncut [16] version of spectral clustering and the results were slightly worse than those obtained by the NJW algorithm. We also ran the k-means algorithm (using the i-th row of the weight matrix W as the feature vector for the point i) and its results were the worst. It can be seen that on most datasets ITPC outperformed the other methods or at least produced quite similar results which indicates that the MI clustering score is more suitable for pairwise clustering than the Ncut score. Note that the Graclus algorithm outperforms spectral methods which validates our optimization strategy and indicates that direct optimization of a pairwise clustering score is better (and faster) than applying eigen-vector based methods. Note also that in one case the NJW algorithm failed badly (20ngB) and in another case (USPS-17) the PIC algorithm failed badly. The most likely cause being that the top eigen-vectors of the graph Laplacian failed to provide a good low-dimensional embedding for the k-means. Such problem does not exist in sequential optimization.

The ITPC algorithm utilizes a greedy approach to maximize the mutual information score $I(Y_1;Y_2)$ (6). In principle, this optimization approach can get stuck in local maxima points. Next we demonstrated that in the datasets we used there was no problem of getting stuck in local optima. Using the ground-truth labels we can compute the mutual-information score of the true clustering and compare it to the score of the clustering obtained by the ITPC algorithm. Table 3 shows the mutual-information score for all the datasets we used. In all cases the score of the clustering obtained by ITPC algorithm was higher than the score of the true clustering. Therefore, although there is no guarantee that we obtained the global maximum, it indicates that our optimization process works well.

Table 3. Comparison of the cluster-membership mutual-information score (6) of the ITPC clustering vs. the ground-truth clustering.

Dataset	**ITPC Score**	**True Score**
Iris	.949	.903
Glass	1.127	.349
Wine	.806	.761
WDBC	.474	.413
OlFace5	.554	.382
USPS-01	.580	.564
USPS-17	.539	.502
USPS-245	.916	.871
20ngA	.599	.595
20ngB	.535	.530
20ngC	.775	.756
20ngD	.886	.849

Although in pathological cases a sequential algorithm can take many iterations until convergence, in practice the number of needed iterations is much less than the number of points. In our experiments we limited the number of iterations on the data points to be 30. Note that in spectral methods, even if we use efficient algorithms to find eigenvectors, in the second step we apply k-means on the embedding results and we face a complexity issue that is also solved by limiting the number of k-means iterations.

7 Conclusion

To conclude, we introduced a simple pairwise clustering method based on applying a random-walk associated with the affinity matrix of the data points and computing the mutual information between visited clusters. The main point of our paper is defining an information theoretical criterion for pairwise clustering and showing that it yields better results than Ncut criterion and its variants. Dhillon et al. [3] showed that direct optimization of Ncut, using variants of K-means, outperforms spectral methods (that optimize an approximated cost function) in terms of both accuracy and complexity. Hence, even if we try hard to develop efficient spectral clustering variants we will not gain much. We validated this observation in Table 2.

The proposed ITPC method has linear computational complexity which makes it easily scalable for large datasets. Therefore, our algorithm is applicable to large-scale clustering tasks. Experimental results show that our algorithm outperforms state-of-the-art pairwise clustering algorithms in terms of speed, memory usage, and clustering quality. A possible weakness of the greedy method we used for optimization is getting stuck in local optima points. We showed, however, that this problem does not occur in the real datasets we analyzed. The main advantage of spectral clustering is that there is an analytic solution (for a relaxation of the Ncut cost function) and hence there is no problem of getting stuck on local optimum. We can combine ITPC and spectral clustering by first applying spectral clustering on a small subset of our data and using the result as a starting point for our approach by merging the other points to one of the obtained clusters. In this study we concentrated on the problem of pairwise clustering. The proposed method can be applied also to the more general problem of aggregating the states of a large scale Markov chain.

References

1. Chung, F.: Spectral graph theory. CBMS Regional Conference Series in Mathematics, vol. 92 (1997)
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Interscience (1991)
3. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: A multilevel approach. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), pp. 1944–1957 (2007)
4. Dhillon, I., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research 3, 1265–1287 (2003)
5. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: ACM SIGKDD (2003)
6. Dubnov, S., El-Yaniv, R., Gdalyahu, Y., Schneidman, E., Tishby, N., Yona, G.: A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. Macine Learning 47(1), 35–61 (2002)
7. Goldberger, J., Erez, K., Abeles, M.: A Markov clustering method for analyzing movement trajectories. In: IEEE Machine Learning for Signal Processing Workshop, MLSP (2007)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
9. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Scientific Computing, 359–392 (1999)
10. Lin, F., Cohen, W.: Power iteration clustering. In: Int. Conf. on Machine Learning (2010)
11. Manning, C., Raghavan, P., Schutze, H.: Introduction to information retrieval. Cambridge University Press (2008)
12. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: AISTATS (2001)
13. Mitchell, T.: Machine learning. McGraw Hill (1997)

14. Ng, A.Y., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14 (2002)
15. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: IEEE Workshop on Applications of Computer Vision (1994)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. pattern Anal. Machine Intell. 22(8), 888–905 (2000)
17. Slonim, N., Friedman, N., Tishby, N.: Unsupervised document classification using sequential information maximization. In: Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (2002)
18. Slonim, N., Tishby, N.: Document clustering using word clusters via the information bottleneck method. In: Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (2000)
19. Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Allerton Conf. on Communication, Control and Computing (1999)
20. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing, 395–416 (2007)
21. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: Int'l Conf. Computer Vision (2003)
22. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems 17 (2005)